

# Automated segmentation by pixel classification of retinal layers in ophthalmic OCT images

K. A. Vermeer,<sup>1,\*</sup> J. van der Schoot,<sup>1,2</sup> H. G. Lemij,<sup>2</sup> and  
J. F. de Boer<sup>1,3,4</sup>

<sup>1</sup>Rotterdam Ophthalmic Institute, Rotterdam Eye Hospital, P.O. Box 70030, 3000 LM Rotterdam, The Netherlands

<sup>2</sup>Glaucoma Service, Rotterdam Eye Hospital, P.O. Box 70030, 3000 LM Rotterdam, The Netherlands

<sup>3</sup>Dept. of Physics and Astronomy, VU University, De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands

<sup>4</sup>LaserLaB Amsterdam, VU University, De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands

[\\*k.vermeer@eyehospital.nl](mailto:k.vermeer@eyehospital.nl)

**Abstract:** Current OCT devices provide three-dimensional (3D) in-vivo images of the human retina. The resulting very large data sets are difficult to manually assess. Automated segmentation is required to automatically process the data and produce images that are clinically useful and easy to interpret. In this paper, we present a method to segment the retinal layers in these images. Instead of using complex heuristics to define each layer, simple features are defined and machine learning classifiers are trained based on manually labeled examples. When applied to new data, these classifiers produce labels for every pixel. After regularization of the 3D labeled volume to produce a surface, this results in consistent, three-dimensionally segmented layers that match known retinal morphology. Six labels were defined, corresponding to the following layers: Vitreous, retinal nerve fiber layer (RNFL), ganglion cell layer & inner plexiform layer, inner nuclear layer & outer plexiform layer, photoreceptors & retinal pigment epithelium and choroid. For both normal and glaucomatous eyes that were imaged with a Spectralis (Heidelberg Engineering) OCT system, the five resulting interfaces were compared between automatic and manual segmentation. RMS errors for the top and bottom of the retina were between 4 and 6  $\mu\text{m}$ , while the errors for intra-retinal interfaces were between 6 and 15  $\mu\text{m}$ . The resulting total retinal thickness maps corresponded with known retinal morphology. RNFL thickness maps were compared to GDx (Carl Zeiss Meditec) thickness maps. Both maps were mostly consistent but local defects were better visualized in OCT-derived thickness maps.

© 2011 Optical Society of America

**OCIS codes:** (100.0100) Image processing; (100.2960) Image analysis; (100.5010) Pattern recognition; (170.4470) Ophthalmology; (170.4500) Optical coherence tomography; (170.4580) Optical diagnostics for medicine.

## References and links

1. N. Nassif, B. Cense, B. Park, M. Pierce, S. Yun, B. Bouma, G. Tearney, T. Chen, and J. de Boer, "In vivo high-resolution video-rate spectral-domain optical coherence tomography of the human retina and optic nerve," *Opt. Express* **12**, 367–376 (2004).
2. T. Fabritius, S. Makita, M. Miura, R. Myllyla, and Y. Yasuno, "Automated segmentation of the macula by optical coherence tomography," *Opt. Express* **17**, 15659–15669 (2009).
3. A. Mishra, A. Wong, K. Bizheva, and D. A. Clausi, "Intra-retinal layer segmentation in optical coherence tomography images," *Opt. Express* **17**, 23719–23728 (2009).
4. V. Kajić, B. Povazay, B. Hermann, B. Hofer, D. Marshall, P. L. Rosin, and W. Drexler, "Robust segmentation of intraretinal layers in the normal human fovea using a novel statistical model based on texture and shape analysis," *Opt. Express* **18**, 14730–14744 (2010).
5. M. K. Garvin, M. D. Abramoff, X. Wu, S. R. Russell, T. L. Burns, and M. Sonka, "Automated 3-D intraretinal layer segmentation of macular spectral-domain optical coherence tomography images," *IEEE Trans. Med. Imaging* **28**, 1436–1447 (2009).
6. S. J. Chiu, X. T. Li, P. Nicholas, C. A. Toth, J. A. Izatt, and S. Farsiu, "Automatic segmentation of seven retinal layers in SDOCT images congruent with expert manual segmentation," *Opt. Express* **18**, 19413–19428 (2010).
7. M. Mujat, R. Chan, B. Cense, B. Park, C. Joo, T. Akkin, T. Chen, and J. de Boer, "Retinal nerve fiber layer thickness map determined from optical coherence tomography images," *Opt. Express* **13**, 9480–9491 (2005).
8. Q. Yang, C. A. Reisman, Z. Wang, Y. Fukuma, M. Hangai, N. Yoshimura, A. Tomidokoro, M. Araie, A. S. Raza, D. C. Hood, and K. Chan, "Automated layer segmentation of macular OCT images using dual-scale gradient information," *Opt. Express* **18**, 21293–21307 (2010).
9. H. Zhu, D. P. Crabb, P. G. Schlottmann, T. Ho, and D. F. Garway-Heath, "FloatingCanvas: quantification of 3D retinal structures from spectral-domain optical coherence tomography," *Opt. Express* **18**, 24595–24610 (2010).
10. R. J. Zawadzki, A. R. Fuller, D. F. Wiley, B. Hamann, S. S. Choi, and J. S. Werner, "Adaptation of a support vector machine algorithm for segmentation and visualization of retinal structures in volumetric optical coherence tomography data sets," *J. Biomed. Opt.* **12**, 041206 (2007).
11. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE CVPR* **1**, 511–518 (2001).
12. V. N. Vapnik, *The Nature of Statistical Learning Theory* (Springer-Verlag, 1995).
13. V. N. Vapnik, *Estimation of Dependences Based on Empirical Data* (Springer-Verlag, 1982).
14. C. Cortes and V. N. Vapnik, "Support-vector networks," *Mach. Learn.* **20**, 273–297 (1995).
15. M. Aizerman, E. Braverman, and L. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning," *Autom. Rem. Control* **25**, 821–837 (1964).
16. Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J. Lin, "Training and testing low-degree polynomial data mappings via linear SVM," *J. Mach. Learn. Res.* **11**, 1471–1490 (2010).
17. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes - Active Contour Models," *Int. J. Comput. Vision* **1**, 321–331 (1987).
18. S. Osher and N. Paragios, *Geometric Level Set Methods in Imaging, Vision, and Graphics* (Springer-Verlag, 2003).
19. B. Cense, T. C. Chen, B. H. Park, M. C. Pierce, and J. F. de Boer, "Thickness and birefringence of healthy retinal nerve fiber layer tissue measured with polarization-sensitive optical coherence tomography," *Invest. Ophthalmol. Vis. Sci.* **45**, 2606–2612 (2004).
20. X.-R. Huang, H. Bagga, D. S. Greenfield, and R. W. Knighton, "Variation of peripapillary retinal nerve fiber layer birefringence in normal human subjects," *Invest. Ophthalmol. Vis. Sci.* **45**, 3073–3080 (2004).
21. H. Ishikawa, J. Kim, T. R. Friberg, G. Wollstein, L. Kagemann, M. L. Gabriele, K. A. Townsend, K. R. Sung, J. S. Duker, J. G. Fujimoto, and J. S. Schuman, "Three-dimensional optical coherence tomography (3d-oct) image enhancement with segmentation-free contour modeling c-mode," *Invest. Ophthalmol. Vis. Sci.* **50**, 1344–1349 (2009).
22. S. Jiao, R. Knighton, X. Huang, G. Gregori, and C. Puliafito, "Simultaneous acquisition of sectional and fundus ophthalmic images with spectral-domain optical coherence tomography," *Opt. Express* **13**, 444–452 (2005).
23. E. C. Lee, J. F. de Boer, M. Mujat, H. Lim, and S. H. Yun, "In vivo optical frequency domain imaging of human retina and choroid," *Opt. Express* **14**, 4403–4411 (2006).

## 1. Introduction

Current spectral domain Optical Coherence Tomography (OCT), as implemented by various manufacturers for ophthalmic applications, produces high quality data at a high speed. Typically, around 15,000–40,000 A-lines (depth scans at a single location) are produced per second. This high speed enables the acquisition of three-dimensional or volumetric data sets in a short period of time [?]. Combined with some method for motion correction, densely sampled vol-

umes may be produced with increased signal-to-noise ratio and reduced speckle due to averaging, while the acquisition time (including alignment and focusing) is just a few minutes, even in pathological eyes.

Densely sampling a volume results in large data sets (in the order of 50 million pixels) which are difficult to quickly analyze in a clinical setting. Adequately representing three-dimensional data on a display is challenging and the available time for review is very limited. While the data contains a large amount of information, often much of it is irrelevant for the specific task at hand, such as glaucoma detection or monitoring. Reducing the data to a much smaller and easier to interpret set, still containing most relevant information, is therefore vital for routine clinical use. In addition, reducing the data is required for most tasks related to computer-aided diagnosis.

In most cases, segmentation of the data is a prerequisite for performing the mentioned data reduction. In the case of retinal OCT images, this means that the layers of the retina have to be labeled automatically. Several methods for segmentation of OCT data have been described previously and are briefly reviewed below.

Fabritius et al. proposed a straight-forward intensity based method to find only the inner limiting membrane and the retinal pigment epithelium (RPE), providing a three-dimensional segmentation of the retina [?]. Mishra et al. employed a two-step method based on a kernel-based optimization scheme to segment B-scans of rodent retinas [?]. Kajić et al. introduced a modification of an active appearance model for segmenting macular OCT data, which was trained on a large number of manually labeled example images [?]. Garvin et al. introduced graph cuts to localize the boundary between layers, which are guaranteed to find the global optimum [?]. Weights were defined heuristically and shape models were derived from training data. Instead of graph cuts, Chiu et al. used dynamic programming techniques as a more efficient technique to localize various retinal layer interfaces in single B-scans [?].

A significant portion of those employ a segmentation per two-dimensional OCT frame [?, ?], sometimes subsequently combined to a three-dimensional result by post-processing [?, ?, ?], while others offer a true three-dimensional segmentation [?, ?, ?].

In this paper, we present a method for three-dimensional retinal layer segmentation in OCT images by a flexible method that learns from provided examples. Parts of representative OCT scans were manually segmented and used by the algorithms to learn from. Learning and classification of pixels was done by a support vector machine, although many other machine learning classifiers may be employed alternatively. Smoothness of the detected interfaces was guaranteed by the level set regularization that was applied after the pixel classification. The procedure was the same for all layers, except for the manually segmented data used to train the classifier.

Similar to Zawadzki et al. [?], a support vector machine is used to classify pixels in the OCT image, but we aim to do this in a fully automated way. Like in other previous work [?, ?], our method uses example data (manually labeled scans) to learn from. However, we do not put strong restrictions on the shape of the interfaces or layers, because the segmentation should also be applicable to atypical (i.e., diseased) retinas, which shape is not represented by the learning data set.

Results on different interfaces between layers are presented and evaluated for both normal and glaucomatous eyes. Validation was performed by comparing the automatically and manually segmented interfaces. Further processing of the algorithms' outcome produced thickness maps of single or combined layers, which can be used for clinical assessment of the pathology of the imaged eye, and retinal and choroidal blood vessel images.

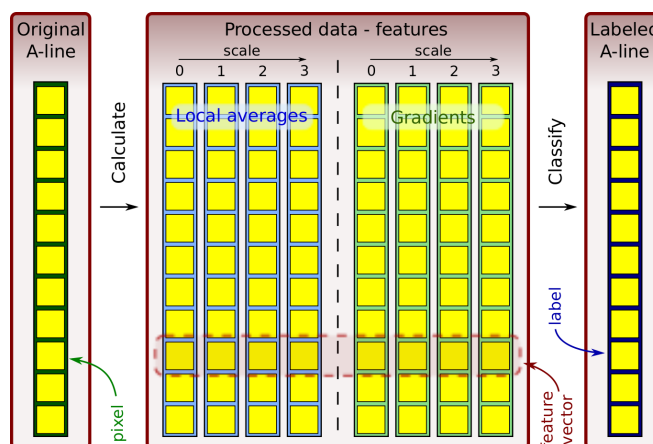


Fig. 1. Overview of the feature calculation and classification process. Every A-line is processed to produce averages and gradients at different scales, thereby transforming every pixel into a feature vector. The classifier calculates the label based on the feature vector, resulting in a labeled A-line.

## 2. Methods

The full algorithm is comprised of three steps: defining features, classifying pixels and performing regularization. Each of these steps are further explained in this section. In the learning stage, the features are defined and the classifier is trained. No regularization is required in this case. An overview of the first two steps is given in Fig. ???. Note that features are defined based on individual A-lines, resulting in a feature vector for each pixel. These pixels are then individually classified and finally the labels in the whole volume are regularized to produce a smooth interface.

When determining multiple interfaces, the described method is applied subsequently to each interface of interest. On the retinal OCT scans, a hierarchical approach is taken: first, the top and bottom of the retina are detected and then the intra-retinal interfaces are localized. These intra-retinal interfaces are forced to lie within the retina. However, the order of intra-retinal interfaces is not enforced.

### 2.1. Features

Classification of pixels is generally done based on one or more features of these pixels. In OCT data, the most basic feature is simply the value produced by the OCT measurement. However, given that a backscatter value is not specific for any tissue, the data cannot be segmented based on only that. For example, both the RNFL and the RPE are strongly backscattering layers in the retina.

Our features are defined based on two observation. First, as explained above, incorporating only the pixel value itself is insufficient. Instead, data from pixels above and below the current one should be incorporated as well. Second, an interface is often delineated by an increase or decrease of the OCT signal, resulting in an intensity edge in the B-scan. We chose to define features based on individual A-lines. This enables the use of the same features (and therefore classifiers) irrespective of the scan protocol (i.e., the number of A-lines per B-scan, or the number of B-scans per volume). Following this reasoning, we used one dimensional Haar-like features [?]. We incorporated averages and gradients, both on different scales. Haar-like features were chosen over, for example, Gaussian averages and differences, because of their

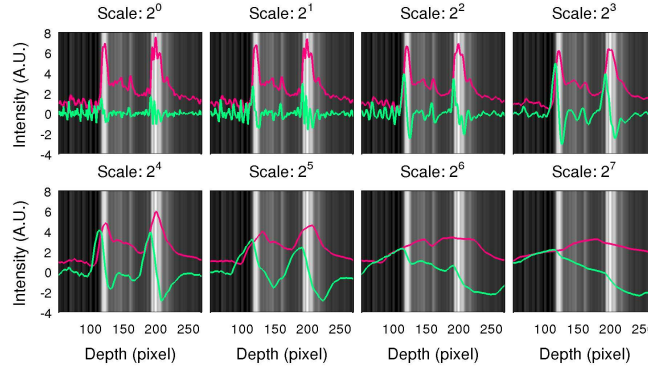


Fig. 2. Features calculated for each pixel on an A-line, at different scales. The A-line is displayed in grayscale in the background of each graph. The features are defined by averages (red lines) and gradients (green lines).

fast implementation by means of lookup tables.

Let the intensity along an A-line be denoted by  $f_{x,y}(z)$ , where  $x$  and  $y$  are the lateral coordinates of the A-line and  $z$  is the depth or distance in axial direction. In the remainder, we will skip the lateral coordinates and simply write  $f(z)$ . Then the first feature,  $g^0$ , is simply  $f$  itself:

$$g^0(z) = f(z). \quad (1)$$

Next, the averages  $g^d$  at scale  $d$  are defined by simply averaging  $2^d$  pixels centered on  $f$

$$g^d(z) = \frac{1}{2^d} \sum_{z'=1-2^{d-1}}^{1+2^{d-1}} f(z+z'). \quad (2)$$

Similarly, the gradient  $h^0$  is calculated by

$$h^0(z) = f(z+1) - f(z) = g^0(z+1) - g^0(z) \quad (3)$$

and the gradients  $h^d$  at scale  $d$  are defined by

$$h^d(z) = g^d(z+2^{d-1}) - g^d(z-2^{d-1}). \quad (4)$$

Based on these features, we define the full feature vector  $\mathbf{x}(z)$  up to scale  $d$  for each pixel as

$$\mathbf{x}(z) = [g^0(z), h^0(z), g^1(z), h^1(z), \dots, g^d(z), h^d(z)]. \quad (5)$$

An example of calculating these features for an A-line is shown in Fig. ???. Similarly, the result for a full B-scan is shown in Fig. ??. In this figure and in the overview of Fig. ??, only 4 scales for both averages and gradients are indicated. However, in our experiments, 8 scales of both types were used (as in Fig. ??).

## 2.2. Classification

A classifier produces a label for each input or feature vector  $\mathbf{x}$ . A specific type of a classifier is the support vector machine (SVM) [?]. During training, it aims at creating a maximum margin between the classification boundary and the samples closest to this boundary [?,?]. When given a new, unlabeled feature vector  $\mathbf{x}$ , the SVM evaluates

$$s(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \quad (6)$$

and uses the sign of  $s(\mathbf{x})$  to produce the label. Here,  $\langle \cdot, \cdot \rangle$  denotes the inner product,  $\mathbf{w}$  denotes the normal of the (linear) classification boundary and  $b$  is some offset. In the training stage,  $\mathbf{w}$  is defined as a weighted sum of the training samples  $\mathbf{x}_i$ . Due to the way SVMs are optimized, many of the weights may go to zero and effectively only a relatively small number of samples, the *support vectors* will be used to define  $\mathbf{w}$ . Equation (??) may now be rewritten as

$$s(\mathbf{x}) = \sum_{i=1}^N \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b, \quad (7)$$

where  $\alpha_i$  denotes the weight of training sample  $\mathbf{x}_i$  and  $y_i$  denotes its corresponding label ( $\pm 1$ ).

The classifier of Eq. (??) is a linear classifier, given that its result is a linear combination of the inner product of the feature vector  $\mathbf{x}$  and the support vectors. However, by replacing the inner product by a kernel  $K(\cdot, \cdot)$  [?], a non-linear SVM is constructed:

$$s(\mathbf{x}) = \sum_{i=1}^N \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b. \quad (8)$$

Implicitly, the kernel maps the input features into a possibly very high dimensional space. In this feature space, a linear classification is performed. Various kernels may be used, such as polynomial kernels or radial basis functions. In the latter case, the kernel maps the input features into an infinite dimensional space, giving highly non-linear classification boundaries. With polynomial kernels, the dimension of the feature space is better controlled.

In general, the kernel-SVM, given by Eq. (??), cannot be rewritten as an explicit linear function as in Eq. (??) or (??). The disadvantage of the implicit form of Eq. (??) is that it requires the storage of all support vectors and, for every new sample, it needs to calculate the kernel for each support vector.

In some cases, however, the kernel can be written explicitly. This applies, for example, to the polynomial kernel  $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$ , where  $d$  is the degree of the polynomial. For higher order polynomial kernels, the resulting mapping results in a highly dimensional feature space, but for lower order kernels (degree 2 and possibly 3), explicit calculation is feasible. This is done by writing the kernel as an inner product of a mapping  $\phi(\cdot)$ :

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle. \quad (9)$$

For example, for a polynomial kernel of degree 1, the corresponding mapping is simply  $\phi(\mathbf{x}) = (1, x_1, \dots, x_n)^T$ , where  $x_i$  is the  $i$ -th element of vector  $\mathbf{x}$ , containing  $n$  elements. In a similar

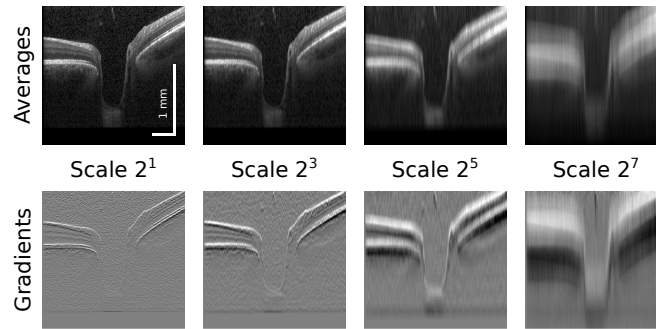


Fig. 3. Graphical representation of the features for each pixel in a B-scan. The feature vector for each pixel consists of one matching pixel from each of the processed B-scans.



way, explicit mappings may be found for polynomial kernels in general [?]. If such an explicit mapping  $\phi(\cdot)$  exists, Eq. (??) may be rewritten as

$$s(\mathbf{x}) = \sum_{i=1}^N \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b = \sum_{i=1}^N \alpha_i y_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle + b = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b, \quad (10)$$

yielding a similar result as Eq. (??). As a result,  $\mathbf{w}$  may now be precomputed and for new data  $\mathbf{x}$  only the mapping  $\phi(\mathbf{x})$  and its inner product with  $\mathbf{w}$  needs to be calculated.

In our application, a polynomial kernel of degree 2 was chosen, with the corresponding mapping

$$\phi(\mathbf{x}) = \left(1, \sqrt{2}x_1, \dots, \sqrt{2}x_n, x_1x_2, \dots, x_{n-1}x_n, x_1^2, \dots, x_n^2\right) \quad (11)$$

which transformed vector  $\mathbf{x}$  from an  $n$ -dimensional space into an  $(n+2)(n+1)/2$ -dimensional space. By precomputing  $\mathbf{w}$ , storing all support vectors was no longer required. In addition, calculation was much faster due to the linear operation of the resulting SVM.

### 2.3. Regularization

The process of pixel classification leads to a volume of pixels with class labels. These labels denote that, according to the classification procedure, the pixel is above or below the interface of interest. The classification result may contain some errors, resulting in incorrectly assigned labels. In addition, imaging artifacts may lead to misclassified A-lines and registration errors result in discontinuities in the interface. Simply using every change in label as an interface would result in a very unrealistic morphology of the layers. Instead, the detected interface will first be regularized by applying some constraints. By penalizing the curvature of the interface, its smoothness can be controlled.

One way of doing this is by using level set methods, which provide a non-parametric way to describe the interface. In contrast with parametric methods, such as snakes [?] that provide an explicit parametrization of the interface, level sets [?] embed the interface implicitly, which has some computational advantages (i.e., regarding propagation and topology of the interface).

The level set function  $\phi$  is defined in the same space as the input data (which is three dimensional for volumetric OCT data) and maps an input coordinate  $\mathbf{x}$  to a scalar. The interface is then defined as the curve  $C$  for which the level set is zero:  $C = \{\mathbf{x} | \phi(\mathbf{x}) = 0\}$ .

The level set is evolved according to the general level set equation

$$\phi_t = -F|\nabla\phi|. \quad (12)$$

Here,  $\phi_t$  is the update step of the level set,  $F$  is some force that drives the level set and  $\nabla\phi$  is the gradient of the level set. Adding the smoothness constraint (based on the curvature  $\kappa$ , which may be calculated directly from the level set function by  $\kappa = \nabla \cdot (\nabla\phi/|\nabla\phi|)$ ) and defining  $F$  by the label field  $L(\mathbf{x})$  results in

$$\phi_t = \alpha\kappa|\nabla\phi| - \beta L(\mathbf{x})|\nabla\phi|, \quad (13)$$

where the ratio of  $\alpha$  and  $\beta$  define the relative contributions of both terms. The label field  $L$  is produced by the classification routine explained in the previous section.

### 2.4. Data

10 healthy subjects and 8 glaucomatous subjects were included from an ongoing study in the Rotterdam Eye Hospital. Of each healthy subject, one eye was selected randomly and included in this study. Subjects with diseased eyes were randomly selected and one moderately glaucomatous eye was included per subject. Moderate glaucoma was defined by a visual field with a

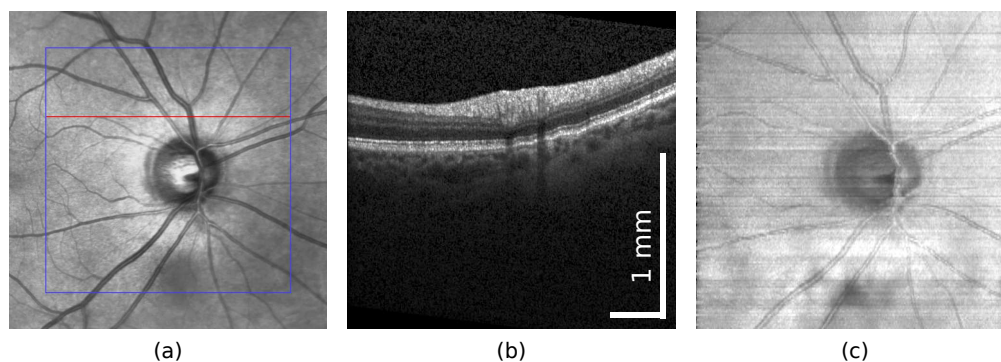


Fig. 4. (a) SLO image of the retina indicating the position of a B-scan (red line) and the total scan area (blue square). (b) OCT B-scan acquired along the red line of Fig. ??(a). (c) Reconstructed *en-face* image based on 193 B-scans.

mean defect of  $-12 - -6$  dB, as tested on a Humphrey Field Analyzer (Carl Zeiss Meditec, Inc., Dublin, CA, USA) with a standard white on white, 24-2 SITA program.

OCT scans of all eyes were acquired with a Spectralis OCT system (Heidelberg Engineering, Dossenheim, Germany) that simultaneously captures a scanner laser ophthalmoscope (SLO) image. The scan protocol used acquired A-lines of 496 pixels. 512 A-lines were combined into a B-scan and the full volumetric scan comprised 193 B-scans. The system employed an eye-tracker and it was set to acquire and average 5 B-scans to improve the signal-to-noise ratio. The field-of-view was  $20 \times 20^\circ$  (corresponding to an area of almost  $6 \times 6$  mm). The resulting volumetric data is rather anisotropic, with a sampling density of  $89 \times 33 \times 259 \text{ mm}^{-1}$  (in x, y, and z-direction respectively, where x and y are lateral directions, and z is the axial direction). This results in a pixel size of approximately  $11 \times 30 \times 3.9 \mu\text{m}$ .

All analyses were carried out on the raw measurements as produced by the OCT device. This means that the data was not logarithmically transformed (as is often done when displaying OCT data) and no additional registration or alignment of the data was performed. (For display purposes, the printed OCT images in this paper were transformed by  $\sqrt[4]{\cdot}$ , according to the manufacturer's recommendation.)

An example of a B-scan with its corresponding location on the simultaneously acquired fundus SLO image is shown in Figs. ??(a) and ??(b). Many of these B-scans are acquired within the outlined area. In Fig. ??(c), the reconstructed *en-face* image of the OCT volume is shown, indicating that the OCT data suffers from various artifacts, such as fluctuating brightness and residual eye movement.

Two B-scans of each healthy subject and one B-scan of each glaucomatous subject were manually segmented using ITK-SNAP (available at <http://www.itksnap.org/>). Every pixel was assigned one of the following labels: Vitreous, RNFL, ganglion cell layer (GCL) & inner plexiform layer (IPL), inner nuclear layer (INL) & outer plexiform layer (OPL), photoreceptors/RPE or choroid. In this way, five interfaces between these structures were defined. At the location of the optic nerve head (ONH), no labeling was performed because in that area the retina's morphology differs from its usual layered arrangement. The ONH was thus excluded from the training data. An example of a manually labeled B-scan, corresponding to the B-scan in Fig. ??(b), is shown in Fig. ?? (left).

The vertical location of the manually segmented B-scans was chosen in the following way. Per hemisphere, five vertical areas were defined: In the ONH, at the edge of the ONH and three areas of increasing distance from the edge of the ONH. The training set contained two B-scans



in each of these ten areas, ensuring that the full sampled area was covered.

### 2.5. Application

One of the applications of OCT is the assessment of glaucoma. In this progressive eye disease, the RNFL deteriorates and consequently its thickness decreases. Viewing and interpreting a full 3D OCT volume is too time-consuming for integrating in the normal clinical routine. Viewing and interpreting a 2D thickness map, however, is much quicker. In addition, it may easily be compared to normative data or previous measurements.

The RNFL thickness map was derived from the segmentation results by measuring the distance along each A-line from the top RNFL interface to the bottom RNFL interface. For each A-line, a single distance measure results, which is then combined into a 2D map for the whole 3D data volume.

RNFL thickness maps were also acquired by the GDx (Carl Zeiss Meditec, Inc, USA), a Scanning Laser Polarimetry (SLP) device. SLP measures the birefringence of the sampled tissue. Since most of the birefringence is attributed to the retinal nerve fiber layer (RNFL), this produces a measure that is related to the thickness of the RNFL. Assuming (incorrectly [?, ?]) that the resulting retardation is proportionally related to the thickness of the tissue, an RNFL thickness map is produced. The resulting SLP thickness maps were visually compared to the OCT thickness maps produced by the presented segmentation method.

## 3. Results

Five interfaces were evaluated, between the vitreous and the RNFL, between the RNFL and the GCL/IPL, between the IPL and the INL, between the OPL and the ONL and between the RPE and the choroid. These interfaces defined six different areas, consisting of one or more tissue layers.

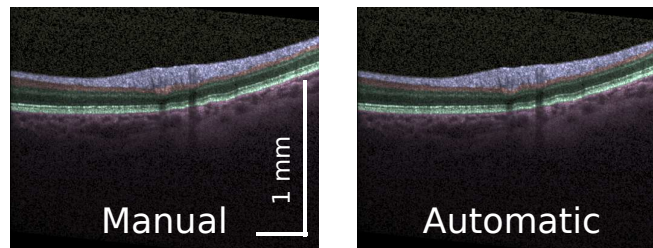


Fig. 5. Manually (*left*) and automatically (*right*) labeled B-scan of a normal eye. This eye was excluded from the training data when obtaining the automatic segmentation.

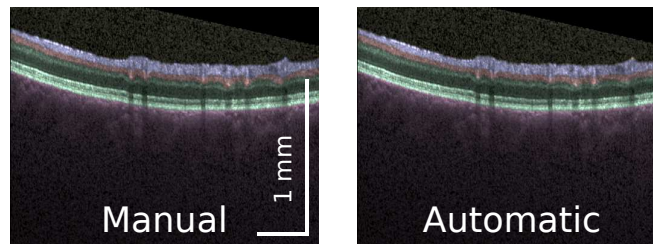


Fig. 6. Manually (*left*) and automatically (*right*) labeled B-scan of a glaucomatous eye. No glaucomatous eye was included in the training data.

Table 1. Localization Errors of Automatically Segmented Interfaces Compared to Manual Segmentations\*

Interface			Root-mean-square error ( $\mu\text{m}$ )		Mean absolute deviation ( $\mu\text{m}$ )	
			Normal	Glaucoma	Normal	Glaucoma
Vitreous	-	RNFL	3.7	5.5	2.7	4.5
RNFL	-	GCL	15.4	11.5	9.1	8.3
IPL	-	INL	15.0	9.5	9.2	6.4
OPL	-	ONL	9.3	5.8	5.8	4.5
RPE	-	Choroid	5.5	6.2	4.2	4.7

\*Error estimates for normal eyes are based on cross-validation.

Because the ONH was excluded from the training data, the results of the automatic segmentation at the ONH's position are invalid. In all numeric evaluations of the algorithm, the area of the ONH was excluded. However, in the displayed figures, the papillary area is not masked out.

An example of the automatic segmentation result on a normal eye is shown in Fig. ??, next to the corresponding manually labeled B-scan. Examples of a glaucomatous eye are shown in Fig. ??.

### 3.1. Accuracy

For each interface, the accuracy was determined by comparing the results of the automatic algorithm with the manually segmented data. Regularization in our method works in 3D, while manual segmentation was done on a single B-scan only. Because B-scans were not realigned (in depth or laterally) before analysis, misaligned B-scans did not match with the automatically segmented interfaces due to the restrictions placed on the interfaces' 3D shapes by the regularization procedure. Manual segmentation, on the other hand, did not incorporate data of adjacent B-scans. Alignment errors of B-scans within the full 3D scan could thus have resulted in increased reported errors.

An overview of the results of the automatic segmentation is shown in Table ?. For evaluation of the errors on the scans of normal subjects, a leave-one-out cross-validation was performed. Considering all B-scans independently in the cross-validation procedure would introduce a bias, due to the correlation of B-scans of the same eye. Instead, cross-validation was implemented by repeatedly removing *both* scans of one eye from the training data set, evaluating the result on those B-scans of the excluded eye and finally averaging the resulting error over all 10 repetitions. Estimation of the error on the glaucomatous eyes was done by training the algorithm on all normal eyes and using the glaucomatous eyes only for error assessment.

For comparison with other methods, two error measures were evaluated: the root-mean-square error and the mean absolute deviation. For our data, an error of  $3.9\mu\text{m}$  corresponded to an axial error of one pixel.

### 3.2. Processing time

The processing times for classification (including calculation of the features) and regularization were recorded on a Intel® Core™ 2 CPU (Q6600), running at 2.4 GHz and containing 4 GB of memory. While this is a multi-core processor, the calculations were not parallelized. The routines were implemented in Matlab (Mathworks, Natick, MA, USA) and used the SVM and level set methods from LIBLINEAR (a library for large linear classification, available at <http://www.csie.ntu.edu.tw/~cjlin/liblinear/>) and ITK (Insight Segmentation and Registration Toolkit, available at <http://www.itk.org/>). Time spent on load-

ing the data sets and some other minor overhead was excluded. The processing times are listed in Table ???. For the three inner retinal interfaces, the reported times do not include the processing time for the outer interfaces.

### 3.3. Thickness maps

Thickness maps were produced for the full retina to visually validate their correspondence to the known retinal morphology. Thickness maps of only the RNFL, useful for glaucoma diagnosis, were produced as well. These thickness maps are shown in Fig. ?? for one healthy and two glaucomatous eyes. For reference, the *en-face* OCT image is shown in the first column of the figure. Finally, thickness maps produced by a GDx are shown as well.

The retinal and RNFL thickness of the healthy eye matches known morphology. The retina is thick near the ONH and also somewhat thicker towards the macula. The RNFL is thicker superiorly and inferiorly compared to nasally and temporally, which is also reflected in the total retinal thickness. In the two glaucomatous cases, the RNFL is thinner than in the normal eye. Local defects may be found in the temporo-superior and temporo-inferior regions for the first glaucomatous case and temporo-inferiorly for the second glaucomatous case. Compared to the GDx images, these defects are better visualized in the OCT-derived thickness maps.

### 3.4. Vessel visualization

For visualization of vessels, the OCT data was averaged over a small region. Instead of interactively, as in the C-mode images by Ishikawa et al. [?], the region was defined automatically based on the segmentation results. Defining the region as a few pixels above the RPE/choroid interface, the retinal vessels were visualized very well due to their strong shading (see Fig. ??), similar to Jiao et al [?]. Likewise, averaging the OCT data over a small area below the RPE/choroid interface showed a pattern that resembles the choroidal vasculature, including shading of the retinal vessels (see Fig. ??), which was done earlier by Lee et al. [?].

## 4. Discussion

We have presented a method to automatically label retinal layers in OCT images. Instead of relying on heuristics to define the boundaries of the layers, a pixel classification was used. The classifiers were trained on manually segmented data and were based on simple Haar-like features. The classifier's performance clearly depends on the type of features that are provided. However, the classifier decides (based on the training data) *which* features are selected and *how* they are combined. For that process, existing techniques for feature selection and classification from the pattern recognition and machine learning literature may be applied.

The presented method is very flexible in that it can easily be adapted to other interfaces. For

Table 2. Processing Times (Standard Deviation) of the Classification (Including Calculating the Features) and Regularization Steps

Interface			Classification time (s)		Regularization time (s)	
			Normal	Glaucoma	Normal	Glaucoma
Vitreous	-	RNFL	82.0 (1.9)	82.7 (2.0)	91 (11)	112 (11)
RNFL	-	GCL	86.6 (1.4)	84.7 (0.7)	95 (17)	104 (23)
IPL	-	INL	74.1 (1.4)	73.7 (1.4)	122 (27)	143 (33)
OPL	-	ONL	74.0 (1.3)	74.3 (1.7)	111 (24)	103 (27)
RPE	-	Choroid	82.8 (1.4)	82.4(1.7)	104 (16)	101 (15)

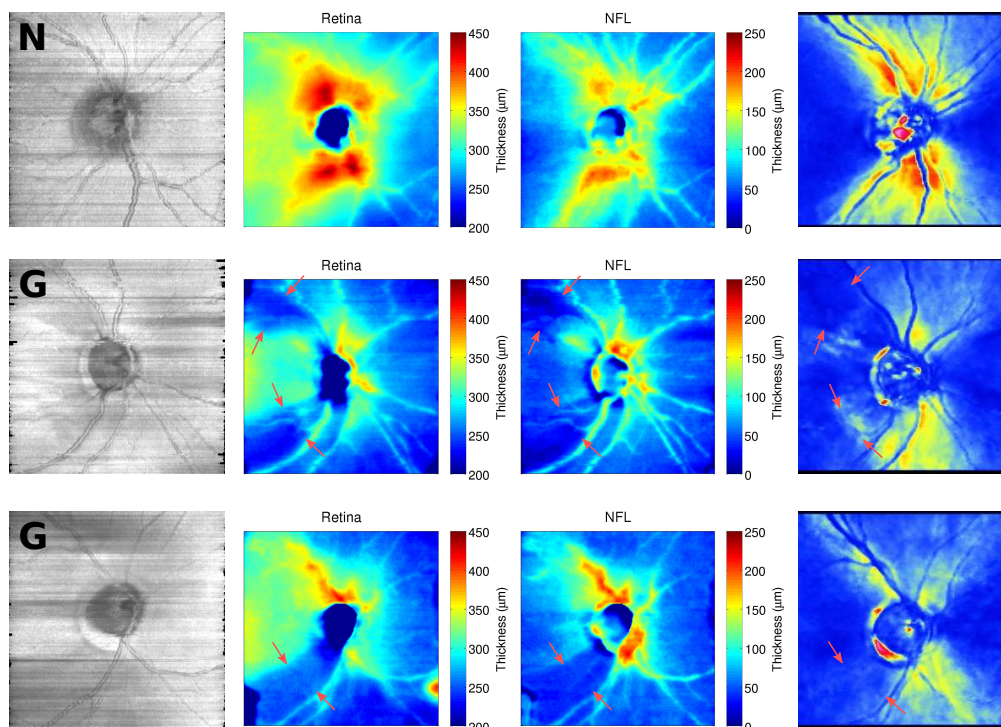


Fig. 7. Thickness maps produced after segmentation. The top row shows the results for a normal eye; the second and third row show the results for glaucomatous eyes. The first column shows the en-face reconstruction, the second column shows the full retinal thickness, the third column shows the RNFL thickness and the last column shows the thickness as assessed by a GDx (dark, blue colors correspond to a thin RNFL and warm, red colors correspond to a thick RNFL). Edges of local defects are indicated by red arrows.

this, the set of manually segmented data has to be extended to include the additional layers and classifiers have to be trained on this data. Then, the new classifiers can simply be applied to the data to obtain an automated segmentation of those layers. Likewise, the method may be extended to other OCT devices. Once a set of manually labeled images acquired on such a device is available, the classifiers may be trained on this data and subsequently applied to new data sets originating from these machines.

The type of features and the type of classifier that was used was not optimized for each interface separately. This paper illustrates how a single approach can be suitable for detecting all interfaces, despite their different features. However, optimizing the choice of features and classifiers per interface is likely to further improve the results. Additionally, the smoothness may be optimized for each interface by adapting the weights  $\alpha$  and  $\beta$  of the level set method in Eq. (??). All of these refinements may also be used to incorporate additional prior knowledge.

For the intra-retinal interfaces, the errors in glaucomatous eyes were actually smaller than those in normal eyes. Because the error on normal eyes was calculated by a leave-one-eye-out method, the available training set when evaluating normal eyes contained less data (i.e., 18 B-scans of 9 eyes) than the training set for evaluation of glaucomatous eyes (i.e., 20 B-scans of 10 eyes). This suggests that the estimated error for normal eyes may have been biased and that the actual generalization error of the full training set is lower than the leave-one-out estimate presented here. In addition, these results imply that increasing the pool of training data, i.e.,

manually segmenting B-scans from more normal eyes, will further improve the results for those interfaces. This error behavior was not observed for the top and bottom interfaces of the retina, which is consistent with the observation that those interfaces are easier to detect and therefore less training data was required for accurate segmentation.

The root-mean-square error was, as expected, always larger than the mean absolute deviation, because it includes both the variance and the bias of the error. However, the ratio of both errors was not very constant. It varied by interface, which means that in some interfaces, the relative variance in the error (i.e., the variance over the squared mean) was larger than in other interfaces. This might have been caused by the automatic algorithm sometimes ‘locking’ to the wrong boundary for those interfaces. Again, this was observed mostly for the intra-retinal layers and adding more training data may improve this.

The running time of the algorithm was dominated by two parts: Pixel classification and interface regularization. Pixel classification is a prime candidate for speed up by parallel computing. Every A-line is processed independently and even a naïve parallelization will therefore result in a very large speed-up. In addition, after the features are calculated from the A-lines, all pixels on that A-line may be processed in parallel as well. Current graphics processing units (GPUs) seem very suitable for this task. Parallelizing the level set method seems less straight-forward. However, the level set method itself is not an integral part of the presented algorithm, but only used to post-process the classification results. Therefore, any method that results in a regularized interface suffices.

In this paper we introduced a segmentation method that learns from example data and may therefore be used to segment various layers in OCT data. The method is not limited to a single

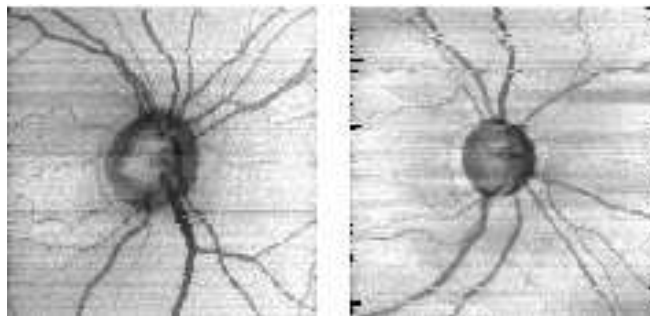


Fig. 8. Integrated OCT data just above the RPE, showing (shadows of) retinal vessels, for a normal (*left*) and a glaucomatous (*right*) eye.

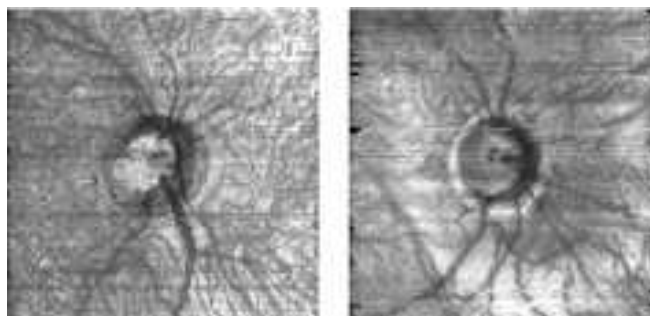


Fig. 9. Integrated OCT data just below the RPE, showing choroidal vasculature (and remnants of retinal vessels), for a normal (*left*) and a glaucomatous (*right*) eye.



OCT device or scan protocol but can adapt to different input data by using new training data. The algorithm first processes single A-lines which are then combined in a three-dimensional regularization procedure, reducing scanning artifacts and resulting in interfaces that appear to match known morphology.

To illustrate the use of the segmentation results, two applications were shown in this paper: thickness maps and vessel projections. NFL thickness maps, such as the ones presented in Fig. ??, are a clinically very useful tool for assessment and monitoring of glaucoma. In a similar way, other thickness maps, representing the thickness of other (combinations of) layers, may be produced as well. In addition, the segmentation results may be used to selectively integrate the OCT signal over specific layers, thus producing an image that shows the scattering properties of those single layers. This may provide additional information on (the progression of) the state of a disease. Finally, the segmentation results may also be used for visualization of the raw OCT data, e.g. by coloring each layer differently. In all of these cases, the described segmentation method is an essential step in transforming a raw OCT scan into a clinically useful representation of the data.

### **Acknowledgments**

This work was sponsored in part by The Rotterdam Eye Hospital Research Foundation, Rotterdam, The Netherlands; Stichting Oogfonds Nederland, Utrecht, The Netherlands; Glaucoomfonds, Utrecht, The Netherlands; Landelijke Stichting voor Blinden en Slechtzienden, Utrecht, The Netherlands; Stichting voor Ooglijders, Rotterdam, The Netherlands; Stichting Nederlands Oogheelkundig Onderzoek, Nijmegen, The Netherlands.